UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF VIRGINIA

SONY MUSIC ENTERTAINMENT, *et al.*, *Plaintiffs*, v. COX COMMUNICATIONS, INC., *et al.*, *Defendants.*	Case No. 1:18-cv-00950-LO-JFA

DECLARATION OF DR. NICK FEAMSTER IN SUPPORT OF COX'S OPPOSITION TO PLAINTIFFS' MOTION FOR SUMMARY JUDGMENT

I, Dr. Nick Feamster, declare as follows:

1. I am the Director of the Center for Data and Computing and Neubauer Professor in the Department of Computer Science at the University of Chicago. I submit this declaration in support of Cox's Memorandum in Opposition to Plaintiffs' Motion for Summary Judgment. I have personal knowledge of the facts stated in this Declaration, and if called as a witness, could testify competently to the matters contained herein.

2. I have reviewed and analyzed the documents produced in this case by Plaintiffs and third parties, including the copyright infringement notices directed to Cox that Plaintiffs produced; Plaintiffs' summaries of the data in the notices (produced as Plaintiffs_00286430); and the MarkMonitor "Evidence Packages" (also referred to as "Cases") produced at MM000306.

3. I have reviewed the Declaration of MarkMonitor's Sam Bahun that was filed in connection with Plaintiffs' Motion for Summary Judgment. Mr. Bahun incorrectly states:

> Two files that have the same hash are identical to one other; they are copies of the same file and have the same contents. Two files that differ in even the smallest way will yield different hash values.

Bahun Summary Judgment Decl. at ¶ 10. Elsewhere in his Declaration Mr. Bahun concedes that this is inaccurate, but only in a footnote to a different paragraph—in the body of which he again states (incorrectly) that "if two files with the same hash are found, the contents of the two files are guaranteed to be identical[.]" *Id.* ¶ 11; *see id.* ¶ 11 & n.1 ("this guarantee is actually short of 100% at a technical matter…."). As a matter of elementary mathematics, given that SHA-1 produces only a 160-bit hash value, but the input to the SHA-1 hash function can be arbitrarily long, for at least some hash values, there must be an infinite number of messages that will hash to the identical value. The same applies to the MD4 and SHA-1 Base 32 hash functions. Indeed, for all of these hash functions, there are an infinite number of messages that will "hash" to each possible hash value. Mr. Bahun's statement is thus incorrect as a general matter.

4. The SHA-1 hash function has been "broken" since 2005, when researchers published a paper demonstrating that for a given SHA-1 hash, it was possible to generate different files with identical hash values (known as "collisions") relatively efficiently.[1] The MD4 hash function has been "broken," in the same sense, for much longer. On March 15, 2006, the National Institute of Standards and Technology (NIST) issued a policy statement that Federal agencies "should stop using SHA-1 for digital signatures, digital time stamping and other applications that require collision resistance as soon as practical."[2]

[1] *See, e.g.*, Bruce Schneier, "Cryptanalysis of SHA-1," Schneier on Security (Feb. 18, 2005), https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html.

[2] https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions.

5. I understand that Plaintiffs claim to have caused notices to be sent to Cox based on activity that MarkMonitor purportedly observed on peer-to-peer networks involving four different peer-to-peer protocols, namely BitTorrent, Ares, eDonkey, and Gnutella. I understand that Plaintiffs have represented that notices corresponding to the different peer-to-peer protocols apply different hash functions, and may use different inputs, according to the following conventions:

a) The hash value provided for BitTorrent protocol notices is an Info Hash. The Info Hash is a SHA-1 hash of certain information contained in a dot-torrent metadata file, which is a type of file that can be used by a BitTorrent client application to join an *ad hoc* BitTorrent network (comprising a number of "peer" computers) and, in theory, retrieve pieces of a content file from the various peers.

b) The hash value provided for Ares protocol notices is a SHA-1 hash of a content file.

c) The hash value provided for Gnutella protocol notices is a SHA-1 base 32 hash of a content file. (Based on my review, the hash type of SHA-1 base 32 hash values is identified in Plaintiffs' notices as simply "SHA-1.")

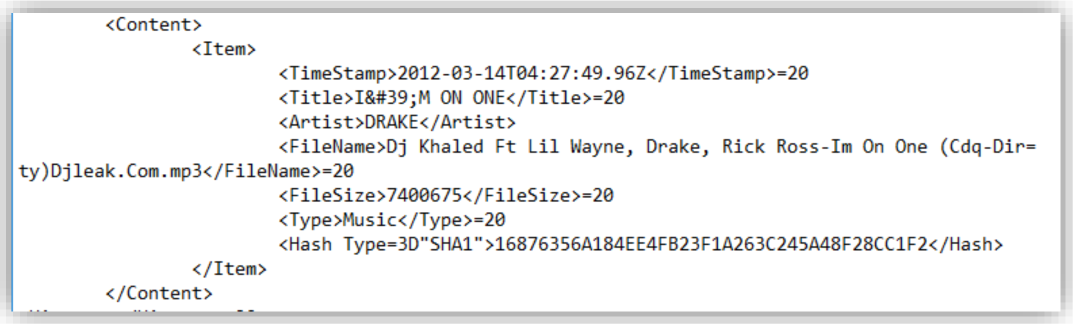
d) The hash value provided for eDonkey protocol notices is an MD4 hash of a content file.

6. In addition to Plaintiffs' summaries of the data in their notices (produced as Plaintiffs_00286430), I have reviewed representative notices that were produced by Plaintiffs in this action. Each of these notices provides a field labeled "<Hash Type>" that also identifies a particular hash function, in this case either SHA-1 or MD4. The <Hash Type> field is followed by a text string (a "hash value") that was presumably created by applying the identified hash function to some file. The notices I reviewed are attached as:

- **Exhibit F-1** (Bates numbered Plaintiffs_00023875, a BitTorrent protocol notice);
- **Exhibit F-2** (Bates numbered Plaintiffs_00053141, an Ares protocol notice);

- **Exhibit F-3** (Bates numbered Plaintiffs_00008847, a Gnutella protocol notice); and
- **Exhibit F-4** (Bates numbered Plaintiffs_00127436, an eDonkey protocol notice).

By way of example, a portion of **Exhibit F-1** (which is a BitTorrent protocol notice and uses the SHA-1 hash) is pictured here:



```
        <Content>
                <Item>
                        <TimeStamp>2012-03-14T04:27:49.96Z</TimeStamp>=20
                        <Title>I&#39;M ON ONE</Title>=20
                        <Artist>DRAKE</Artist>
                        <FileName>Dj Khaled Ft Lil Wayne, Drake, Rick Ross-Im On One (Cdq-Dir=
ty)Djleak.Com.mp3</FileName>=20
                        <FileSize>7400675</FileSize>=20
                        <Type>Music</Type>=20
                        <Hash Type=3D"SHA1">16876356A184EE4FB23F1A263C245A48F28CC1F2</Hash>
                </Item>
        </Content>
```

7. Neither Plaintiffs' summary of notice data, nor the representative notices I have reviewed, contain any indication of what file was used as an input to generate the hash value.

8. SHA-1 (including the SHA-1 base 32 hash and MD4) are classified as "cryptographic" hash functions. Among other things, cryptographic hash functions are "one-way" functions, meaning that it is computationally infeasible to determine the *input* (here, a content file or dot-torrent file) from the *output* (here, the hash value). And because two different inputs can generate the same hash value, it is *impossible* to determine, given a hash value, what input was used to create it.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed this 24th day of September, 2019 in Chicago, Illinois.

/s/

Dr. Nick Feamster